IN THE CLAIMS:

Please amend the claims as follows.

[c1] (Currently Amended) A computer system comprising:

a processor;

~~an object~~ a cache operatively connected to the processor, wherein the cache is configured to operate as both an object cache and a conventional cache using an extended address encoding procedure;

a memory; and

a translator interposed between the ~~object~~ cache and the memory, wherein the translator maps an ~~object~~ encoded address to a physical address within the memory,

wherein the encoded address corresponds to an address comprising an object address embedded in an unused part of a physical address range of the address.

[c2] (Original) The computer system of claim 1, wherein the object address comprises an object identification number and an offset.

[c3] (Cancelled)

[c4] (Cancelled)

[c5] (Currently Amended) The computer system of claim 1, wherein the translator maps the encoded ~~object~~ address to the physical address using a table.

[c6] (Original) The computer system of claim 1, wherein the processor uses an extended instruction set.

[c7] (Currently Amended) The computer system of claim 1, wherein the translator converts operations using the encoded ~~object~~ address into operations using the physical address.

[c8] (Original) The computer system of claim 1, wherein the translator loads a plurality of cache lines from the memory.

[c9] (Original) The computer system of claim 1, wherein the translator allocates physical memory a first time a portion of an object is evicted from the object cache.

[c10] (Currently Amended) A computer system comprising:

a plurality of processors;

~~an object~~ a cache operatively connected to the plurality of processors, wherein the cache is configured to operate as both an object cache and a conventional cache using an extended address encoding procedure;

a memory; and

a translator interposed between the ~~object~~ cache and the memory, wherein the translator maps an ~~object~~ encoded address to a physical address within the memory,

wherein the encoded address corresponds to an address comprising an object address embedded in an unused part of a physical address range of the address.

[c11] (Original) The computer system of claim 10, wherein the object address comprises an object identification number and an offset.

[c12] (Cancelled)

[c13] (Cancelled)

[c14] (Currently Amended) The computer system of claim 10, wherein the translator maps the encoded ~~object~~ address to the physical address using a table.

[c15] (Original) The computer system of claim 10, wherein the plurality of processors uses an extended instruction set.

[c16] (Currently Amended) The computer system of claim 10, wherein the translator converts operations using the encoded ~~object~~ address to operations using the physical address.

[c17] (Original) The computer system of claim 10, wherein the translator loads a plurality of cache lines from the memory.

[c18] (Original) The computer system of claim 10, wherein the translator allocates physical memory a first time a portion of an object is evicted from the object cache.

[c19] (Currently Amended) A method for retrieving an object in a single processor environment comprising:

obtaining an ~~object~~ encoded address corresponding to the object;

determining if the ~~object~~ encoded address corresponds to a tag in a tag array of a cache;

retrieving the ~~object~~ encoded address if the tag corresponding to the object address is in the tag array;

translating the ~~object~~ encoded address into a physical address if the object address is not in the tag array; and

retrieving a cache line using the physical address if the ~~object~~ encoded address is not in the tag array,

wherein the encoded address corresponds to an address comprising an object address embedded in an unused part of a physical address range of the address.

[c20] (Original) The method of claim 19, further comprising:
entering the cache line into the cache.

[c21] (Original) The method of claim 19, wherein the object address comprises an object identification number and an offset.

[c22] (Original) The method of claim 21, wherein a word within the object is retrieved using the offset.

[c23] (Cancelled)

[c24] (Cancelled)

[c25] (Currently Amended) The method of claim 19, wherein the step of translating the ~~object~~ encoded address uses a translator.

[c26] (Currently Amended) The method of claim 25, wherein the translator converts operations using the ~~object~~ encoded address to operations using the physical address.

[c27] (Currently Amended) The method of claim 25, wherein the translator loads a plurality of cache lines from the memory.

[c28] (Currently Amended) A method for retrieving an object in a multiprocessor environment comprising:
obtaining an ~~object~~ encoded address corresponding to the object;
determining if the ~~object~~ encoded address corresponds to a tag in a tag array of a cache;

retrieving the ~~object~~ encoded address if the tag corresponding to the object address is in the tag array;

translating the ~~object~~ encoded address into a physical address if the object address is not in the tag array; and

retrieving a cache line using the physical address if the ~~object~~ encoded address is not in the tag array,

wherein the encoded address corresponds to an address comprising an object address embedded in an unused part of a physical address range of the address.

[c29] (Original) The method of claim 28, further comprising:
entering the cache line into the cache.

[c30] (Original) The method of claim 28, wherein the object address comprises an object identification number and an offset.

[c31] (Original) The method of claim 30, wherein a word within the object is retrieved using the offset.

[c32] (Cancelled)

[c33] (Cancelled)

[c34] (Currently Amended) The method of claim 28, wherein the step of translating an ~~object~~ encoded address uses a translator.

[c35] (Currently Amended) The method of claim 34, wherein the translator converts operations using the ~~object~~ encoded address to operations using the physical address.

[c36] (Original) The method of claim 34, wherein the translator loads a plurality of cache lines from the memory.